Turning the Bucket of Text into a Pipe

Elizabeth G. Hetzler, Vernon L. Crow, Deborah A. Payne, Alan E. Turner

Pacific Northwest National Laboratory[1]

ABSTRACT

Many visual analysis tools operate on a fixed set of data. However, professional information analysts follow issues over a period of time and need to be able to easily add new documents to an ongoing exploration. Some analysts handle documents in a moving window of time, with new documents constantly added and old ones aging out. This paper describes both the user interaction and the technical implementation approach for a visual analysis system designed to support constantly evolving text collections.

CR Categories and Subject Descriptors: I.3.6 [Computer Graphics]: Methodology and Techniques – Interaction techniques, I.6.9 [Visualization] – Information Visualization, Visualization Techniques and Methodologies

Additional Keywords: Information Visualization, Dynamic visualization, User interaction design, real-time updating

1 INTRODUCTION

Many text analysis tools operate on a fixed set of data. In some cases, a fixed set is appropriate, e.g., for common evaluation or for duplication of results. However, this can lead to a focus on "bucket of data" approaches. Professional information analysts follow issues over a period of time. To help them collect and track these ongoing issues, they set up profiles or standing queries that constantly reflect the latest information available on that topic. A search tomorrow will yield additional documents—perhaps only a handful, perhaps hundreds more than today's search. Once the data have been collected, users could benefit from a visual analysis system that allows them to easily add the new documents to an ongoing exploration. If the visualization must be recomputed each time, they lose the context and exploration results stored from earlier work. If users must exit the visualization tool before new computation can take place, then they will be unable to compare differences between the old and new visualizations. In addition, users' perspectives on the issue may change over time, meaning their search criteria will evolve and the decision of which documents are relevant will likewise change. This paper describes both the user interaction and the technical implementation approach for a system designed to support analysis of constantly evolving text collections.

Our initial users are professional information analysts with significant experience using the IN-SPIRE™ visual analytics system. IN-SPIRE is a powerful text information visualization and analysis system developed by the Pacific Northwest National Laboratory [7]. IN-SPIRE supports the rapid perception of key information characteristics in a collection, navigation through the information space, foraging for critical evidence and patterns, and organizing evidence for reasoning. Our main goal was to apply IN-SPIRE to a dynamic document flow, i.e., a continuous feed of documents such that newly arriving documents are added to an existing visualization. Further, we wanted to impose a moving temporal window on the feed, such that records that have been present in the visualization longer than a given time will age off and be removed from the visualization.

Section 2 identifies related work and how this paper differs from it. Section 3 describes the user interaction goals and the design features to achieve these goals. Section 4 summarizes the technical approach that underlies the work. Section 5 describes the user evaluation stages and how they influenced the work. Section 6 provides conclusions and suggests future work.

2 RELATED WORK

Several papers have been written on the visual depiction of change in text datasets, although typically the full set is calculated first, and then change within it is identified. For example, Brandes and Corman [3] describe a visual depiction of verbal discourse networks using translucent layers, although they calculate the layout from a complete static set. Visual change portrayal in dot plots and landscapes is described in [8]. Chi and Card [4] describe visual depictions and interactions of web site structure and usage over time, where each time slot is shown as a visual spreadsheet entry. Galaxy of News [9], the online NewsMap site (http://www.marumushi.com/apps/newsmap/index.cfm) based on the TreeMap metaphor, and Breaking Story [6] visualize news and change in news, although again the visualizations themselves do not appear to be incrementally computed.

This paper differs from this previous work in that it concentrates on the visual discourse of an analyst using a dynamic moving window of documents, calculated incrementally.

In the text analysis area, the Defense Advanced Research Projects Agency (DARPA) Topic Detection and Tracking (TDT) program led to a number of research approaches for algorithmically analyzing a growing set of news documents to identify and cluster evolving stories and distinguish new ones (e.g., [2, 5]). We differ from this work in two key ways:

1. TDT work relates documents about a particular event or news story. Our goal is to find thematically related documents; in fact, finding related documents across multiple specific events can be extremely useful for our purposes.
2. Our work focuses on visual analysis and user interaction with the data, while TDT approaches are focused on system-provided answers.

Wong et al. [10] present MDS-based visualizations for a moving window of data, but do not explore interactive exploration or control by a user. TextPool [1] visualizes dynamic change among text-theme relationships identified in an evolving window of documents and allows user interaction with the temporal

[1]Operated by Battelle for the U.S. Department of Energy
902 Battelle Boulevard, Richland, WA 99354
beth.hetzler@pnl.gov, vern.crow@pnl.gov
debbie.payne@pnl.gov, alan.turner@pnl.gov

IEEE Symposium on Information Visualization 2005
October 23-25, Minneapolis, MN, USA
0-7803-9464-X/05/$20.00 ©2005 IEEE.

controls and theme relations. TextPool shares some goals with our system, although it focuses on portraying change in theme relations, while this work focuses on providing a range of analysis capabilities in the presence of change.

In summary, what distinguishes our system is that it provides dynamic visualization and full text interaction capabilities on a dynamically evolving text dataset.

3 User Interaction Design

From a user interaction standpoint, the main goal of analyzing dynamic data was broken down into several subgoals:

1. Users need to be able to monitor the visualization and see what's new each time a new increment of data is processed and loaded into the system.
2. The system should minimize the disruption to both the analytic process flow and the interaction flow.
3. As much as possible of the interactive capability of IN-SPIRE should be available for dynamic datasets.
4. The system should provide dynamic update features.

3.1 Seeing What's New and What's Old

Our users' priority is to understand the "now" and how it differs from the recent past. A detailed view of topical and collection evolution is a much lower priority. Hence, we focused first on emphasizing and accessing new information. When new increments have been processed and are available, they are automatically loaded into the system and a new visualization is displayed. New topics may have appeared with the new increment and previous topics may age off with the older documents. The previous visualization is displayed up until the moment the new visualization is loaded. This minimizes the disruption and allows the user to more easily see the changes occurring within the visualization.

Identifying and emphasizing what's new is trickier than might be expected, as there are multiple sources of time information. Each document has one or more fields with time and date information; these are analogous to publication dates for a journal. In addition, we have information on when these documents were received by our system. This date-time may have a significant mismatch to the publication date; for example, a document may be published well before it became available to our system.

To solve this issue, we added the concept of Fresh documents—those that have most recently arrived in our system. A color-coded group is created and automatically updated containing recently added documents. A second group, the Stale group, emphasizes documents that have been in the system the longest and will soon age off. In typical use, these two groups comprise a minority; the large majority of documents are in "midlife" and have no special time designation. Figure 1 shows the Fresh and Stale groups in our Group tool. As shown in Figure 2, the Galaxy dots are colored accordingly; yellow dots indicate the newest documents, while blue ones indicate the oldest ones. In this view, all 11 recent documents (the yellow dots) are on the right side of the Galaxy, in two main areas. Such a concentration of new stories may indicate a major story, as opposed to the more typical spread of documents in varied topics.

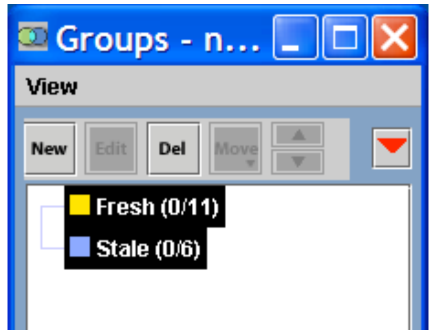

Figure 1. Fresh and stale groups

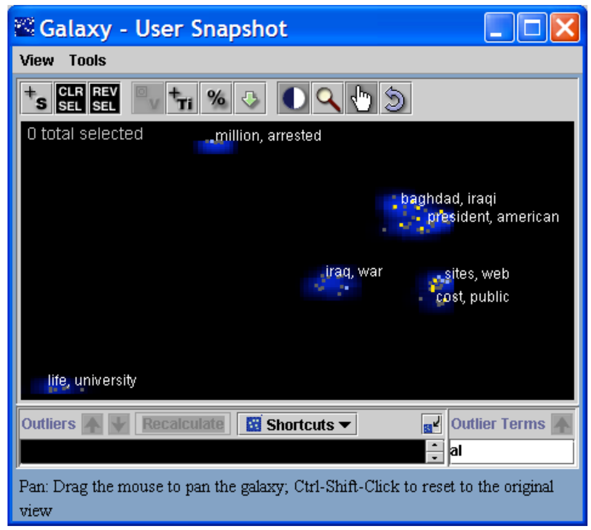

Figure 2. Yellow dots show contexts of newly arrived documents

In contrast, the Time Slicer view uses the publication date-time field to bin documents. This allows the analyst to see the change in publication emphases or see mismatches between publication and availability. In Figure 3, the histogram height shows the number of documents published per hour. The blue ones are about to age off, the yellow ones are new, and a few new ones were actually published a few hours earlier but delayed for some reason in their arrival.

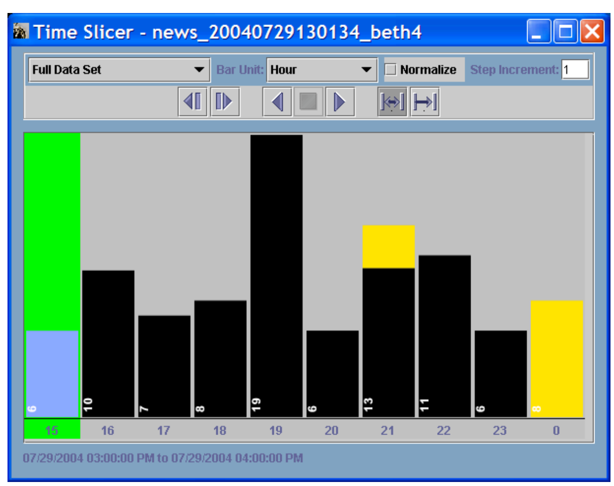

Figure 3. Distribution of fresh (yellow) and stale (blue) documents over time of publication

All of the graphic displays and tools within the IN-SPIRE system are sensitive to the arrival of new increments and the age-off of old documents. Each tool responds as appropriate for that tool. Perhaps the most challenging to implement was the correct aging off of old documents; more details are described in the technical section.

3.2 Controlling Updates and Minimizing Disruption

Users monitoring a dynamic information feed often want new data as soon as possible. Hence, the dynamic system defaults to a "Live" mode, where new increments are loaded as soon as they become available. However, users actively exploring a dataset may not want their view to change in the midst of an interaction. We created a Play/Pause capability to allow users to control the timing of updates (see Figure 4).

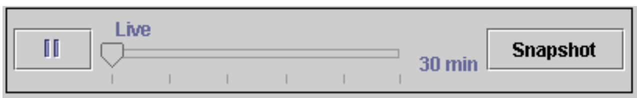

Figure 4. Play/Pause

The user may click the Pause button on the left side of this control, freezing the information space so that new increments will not be loaded. While paused, the Play/Pause panel will turn a cautionary yellow if a new increment becomes available, indicating that the user's current view is out of date (see Figure 5). When the user clicks the Play button, the freeze ends and newer documents will load immediately. Also, the panel displays the number of minutes since the last increment was loaded. In Figure 5, the number 3 indicates that it has been 3 minutes since the last increment was loaded.

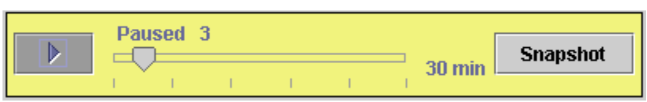

Figure 5. Play/Pause in pause mode

If the current dataset includes a particularly interesting discovery, the user may want to preserve it before accepting new documents. The user can click the Snapshot button on the Play/Pause control to store the current visualization. This action is similar to taking a picture snapshot with a camera but even better, in that the action creates a new dataset that allows complete user interaction. The new snapshot dataset is not connected to the input stream and will not be modified as new data arrive, whereas the original real-time dataset still receives these feeds.

Because IN-SPIRE is intended to support ongoing analysis, we felt it very important to maintain both analytic and interaction states across increments. For example, the analytic state includes the query history, groups made, and documents coded as supporting or refuting hypotheses. Further, an analyst can move documents or terms to the "Outlier area," which effectively lessens their influence on the remaining set; such modifications are also part of the analytic state. These are all maintained. Documents may age out of groups and outliers, but remaining documents retain all memberships and encodings.

To make the interaction even more seamless, we spent considerable time to keep as much as possible of the user's current interaction intact.

- Interaction tools remain open and in the same position.
- Groups remain active and colors show in the Galaxy. Thus a user can highlight the Fresh group to immediately see the locations of new documents and highlight a Key Documents group to immediately see how new information compares to them.
- Documents selected for reading or other interaction remain selected so a user can continue undisturbed.

The only things that might be lost during an incremental update are low-level interactions such as a partial query string, group name, etc. The Play/Pause control is one way to avoid this loss; other options may be explored in future work.

3.3 Providing Full Interactivity

The power of IN-SPIRE is realized through user interaction with the data, not with simple viewing of the visual depictions. Our dynamic version allows users the full set of interactions that are available with regular or non-dynamic datasets. In addition to enabling general exploration and analysis, these capabilities can help a user to focus on recent changes. For example, if an analyst moves all but the most recent documents to the Outlier area, the visualizations temporarily reflect only those new documents (see Figure 6).

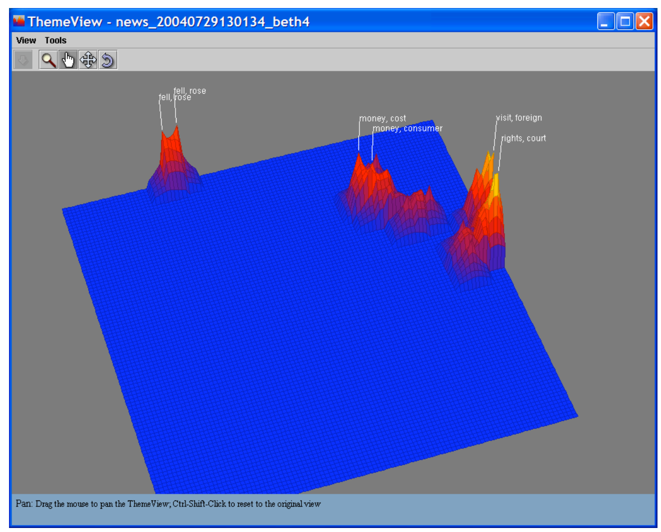

Figure 6. Users can converge on a smaller set, such as the newly arrived documents

An analyst can run queries on the information space, create groups of interesting documents, and see correlations among such groups. Figure 7 shows correlations among the Fresh and Stale groups with queries of interest to the user. In this case, the new documents show a difference in content from the old ones.

Users can also probe themes in the Galaxy, read documents, even create subsets. In short, between increments, a dynamic dataset behaves exactly as a regular one does.

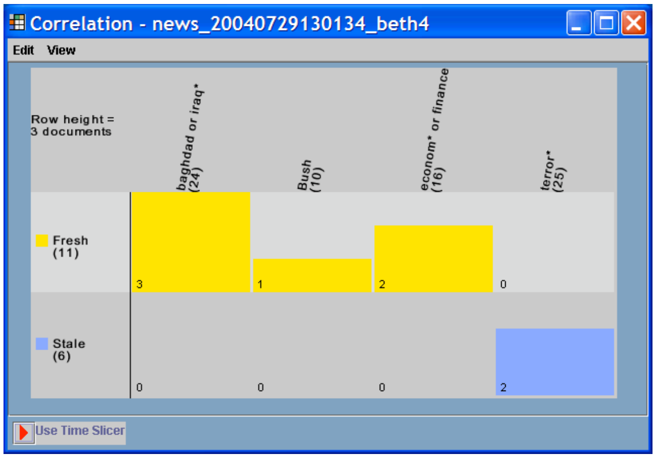

Figure 7. Correlation tool shows overlaps between sets of groups

3.4 Dynamic Update Features

A key user goal is to enable tracking of themes over time. Dynamic clustering is one way to accomplish this goal. For users interested in tracking specific interests, we have a two-prong solution with a powerful advanced user capability and a simpler version for novice or time-stressed users. With our Triage Network capability, users can create queries and query combinations that will be automatically updated as new increments are loaded. Users create such a network by first defining queries, then determining how those queries should be combined, such as unions or intersections. As new documents are added to the dataset, all intermediate and final query combinations are updated. So users can see not only the latest final result, but also the stages along the way. In Figure 8, each box in the leftmost column represents an initial query result. The boxes in successive columns represent combinations of these queries, with the final result shown in the lower right. The colored rectangle in the lower portion of the box shows the relative number of documents retained at each stage. This tool is fully interactive, allowing users to select an intermediate stage to see the document locations in the Galaxy, see how the documents overlap with other stages and groups, and even read the text contents.

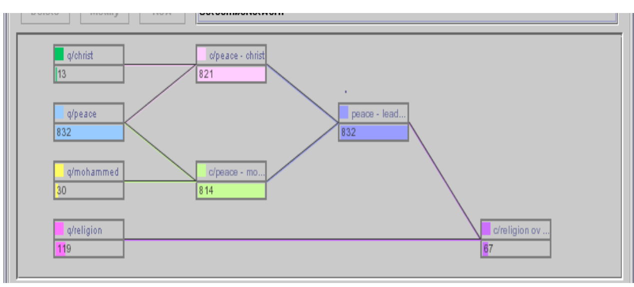

Figure 8. Triage Networks support exploration and updates of specific user interests

Users who want a simpler method for tracking specific interests across updates can simply designate a query result as one to Keep Current. All groups in the Keep Current folder will be recreated by running the original query each time a new increment is loaded. Users can immediately see the changing membership of these groups and see which documents are new.

4 Technical Approach

The dynamic processing goal and the user interaction features described above gave rise to several key design parameters, focusing on performance, support for the Play/Pause capability, and robustness.

4.1 Meeting Performance Goals

Our planning requirements for performance were:

- Maintain temporally moving windows, typically containing 2,000 to 50,000 documents.
- Accommodate small groups of documents (i.e., about 1% of the collection) arriving at varying rates, but typically every 2 to 10 minutes.
- Preserve integrity—the results obtained for processing at any point in the dynamic flow of records should be identical to the results that would be obtained from processing those same records as a static dataset.
- Process a new increment quickly enough, under most circumstances, to allow the user to view the updated visualization before the next increment arrives.

A simplistic approach to the problem of applying IN-SPIRE to a real-time document feed would be to create a complete, new dataset at each increment by removing aged-off records, adding newly arrived records, and then processing the resulting set of records as a static dataset. In fact, this approach was proposed and partially implemented as an interim solution while a better approach was being crafted. This allowed us to validate user needs and gather detailed interaction requirements.

To identify the most profitable areas to target for change, we analyzed the processing stages in creating a dataset and identified these basic steps:

1. Scan the source documents to identify individual records, compile a list of terms (i.e., a vocabulary), and build an index of terms per record.
2. Create an "inverted index" of records per term.
3. Using the index and inverted index, find a subset of N terms that can be used to statistically discriminate among the records.
4. Use the N discriminating terms from step 3 to derive a high-dimensional vector representation of each document.
5. Project the high-dimensional space onto a two-dimensional view, resulting in a visualization.

We found that the scanning and indexing was the time-dominant stage of the process and that optimizing this stage would allow the performance needed to meet the target speed. We crafted a set of algorithms that allowed us to dynamically adjust the contents of the vocabulary and the index, without rebuilding these data structures from scratch. When aged-off records must be removed from the dynamic dataset, we identify terms that are no longer used and remove them from the vocabulary and remove their entries in the index. When new records are added, we add any new terms introduced by these records to the vocabulary and create new entries for the new records in the index. Thus, the vocabulary and index are incrementally adjusted for each new increment. The remaining steps can be performed exactly as they would be for a static dataset.

To test the performance and integrity of the system, we performed a benchmark to compare performance time and visualization results between adding an increment of 100 documents to a dataset of 9,900 and computing a static dataset calculated from the full 10,000 documents. Recall that we defined integrity to mean that the results obtained for processing an increment should be identical to the results that would be obtained from processing the full set of documents as a static dataset. It's not sufficient to compare the appearance of the Galaxy, since the additional documents might have made subtle effects on the visualization, not apparent in the overall appearance. Hence we compared key files, such as document vectors, indices, and dot-plot coordinates. Even small discrepancies in these files would indicate a difference in result. In our experiment, we found the files to be identical between the incremental and the static dataset, so the results of the incremental computation do preserve the integrity of the original.

Adding the 100-document increment to the 9,900 document set and processing it as a dynamic dataset rather than a static 10,000 dataset gave us a time savings of a factor of 20 in the critical scanning stage. The time saved in this particular experiment comes from the fact that scanning, vocabulary building, and indexing runs in linear time with regard to the size and number of records scanned, so by eliminating the need to re-scan records that were scanned in prior increments, we realize a substantial improvement in performance, without sacrificing the accuracy of our results.

4.2 Aging Out Old Documents

One of the most challenging aspects of this work is that not only are new documents added to the set but also old ones must be removed to correctly represent the moving window of data interesting to our users. This requirement affects all data structures that refer to documents, both computational and user interaction structures. Examples include the members of user-created groups, documents marked as "read" and documents in the current selection set; all of these are updated and correctly remapped so that a user will see these maintained across increments. Even documents denoted as "outliers" are remapped and maintained in that state until either the user moves them back into the Galaxy or they age off.

4.3 Support for the User Play/Pause Capability

If a user has paused updates, we needed to process arriving increments in the background so the visualization can be brought up to date as soon as the user releases the pause condition. This raised several questions on the management of the real-time incremental datasets produced by the process and the synchronization of the production of increments with their presentation to, and interaction by, the user. Our solution was to use individual sub-folders in the dataset folder to persist intermediate results produced at each increment and to employ system-wide mutexes to synchronize access to the increment folders and the dataset files by the ingest process and the user.

Providing support for the arrival of updates in the Play mode also required algorithmic changes. Our initial prototype implementation of presenting a new increment to the user was to close and immediately re-open the (now revised) visualization. User testing clearly showed the importance of maintaining as much of the user interaction and analytic state as possible across increments. To accomplish this, each major tool was enhanced with a method to Update itself on the arrival of new increments. For example, the Group tool updates the display of number of group members (since some may have aged off) and ensures that any active groups remain selected; the Triage tool updates itself by rerunning the active query network to ensure that it presents up to date results; and so on for each tool. The server calculates the new increments and holds them until the client is ready to accept them. The client polls for the availability of new increments (unless the user has put it into the Pause state); once one is available, it triggers these Update methods to run. What the user sees is a slight pause while the tools update, then he or she is able to proceed with little disruption.

4.4 Robustness

This system is designed to operate in a real environment where the pace of feeds may vary. We built in capabilities to ensure graceful degradation of the increment capability. For example, if increment processing falls behind the pace of new feeds, the system operates in catch-up mode. Although the user won't see each intervening increment, none of the information in these increments is lost, and it all contributes to the visualization that the user finally sees when the ingest process has caught up with the real-time feed.

Further, real data are not always clean; data may occasionally contain errors ranging from minor to severe. The IN-SPIRE server is robust to many glitches in fields and formatting. Particularly error-ridden data may cause the computation of a specific increment to fail, but such failure does not invalidate the overall dataset. The ingest process skips the bad increment and resumes updated visualizations with the next good increment that arrives.

5 User Evaluation

We carried out two different user evaluation activities: the first using our low-fidelity implementation to assess general feasibility of the approach and fit to workflow, and the second using a beta version of our more complete implementation to assess detailed user interaction. The successive improvements prompted from these evaluations are noted below.

Because the visual approach we used was different from traditional user tools, our first evaluation was done early on, using a functioning but low-fidelity version. The user interaction included a warning of impending update, groups of recently added data, and a subset of the normal IN-SPIRE interaction features. Under the hood, the tool actually recomputed the entire set, rather than only the new additions. This early version allowed us to demonstrate the basic capability to users to explore the kinds of capabilities and interactions they would value. The test allowed us to verify that the visual analysis system could highlight current features of interest to the users, improve their ability to recognize new data, and allow rapid perception and characterization of the "now." This test led to the design of many features described earlier, such as the Play/Pause control, the Fresh and Stale auto-groups, and the Snapshot capability.

This test also changed our basic model of use. Our initial model was that users would employ the incremental version as a monitor but would not want to do much exploring within it. If they wanted to interact fully with the data, they might leave the monitor up but would also open a full-functionality version alongside. This approach proved more awkward than expected; hence, we switched to a model of a single tool, which combined full-exploration functionality with the ability to accept new increments. We added the Play/Pause control so users could temporarily prevent interruption and a Snapshot capability so they could save a particularly interesting state for later use.

Our second user evaluation helped us refine the details of the user interaction. Our goals were:

- Assess user understanding of the controls specific to dynamic datasets, such as Play/Pause and the Fresh/Stale groups.
- See if users could apply the tool capabilities to a representative problem.

We ran this evaluation with three users, all of whom were familiar with the regular IN-SPIRE system and had used it for analyses in the past.

The evaluation was conducted in two parts, matching the two goals above. In the first, we started the incremental IN-SPIRE and asked the users questions about their interpretation of the new controls to hear initial impressions. Then we explained the functionality and asked questions to see if they could correctly interpret the controls and their use. An example of these questions is "show me the documents that have been in the collection the longest."

In the second part, we asked the users to track a particular theme represented in the evolving collection and identify major events related to that theme. To carry out this task, users needed to use the new controls (Play/Pause and Fresh/Stale) in combination with regular IN-SPIRE tools (e.g., clusters, queries, and document viewer).

This user evaluation provided several key findings.

- Clearly we needed to maintain more of the user interaction state across increments than we had. This was an expected result, strongly confirmed by the test. In our recent version, almost all elements of the interaction state are maintained across increments, as explained earlier.
- The ability for a user to track a theme over time, while quite doable, involved too much manipulation and user intervention. This prompted the addition of the Keep Current capability described above in the User Interaction section.
- The incremental interaction controls were intuitive and usable to carry out the functions. Some detailed refinements were suggested, such as clarifying the relationship between the Pause state and the Fresh group contents. These will be folded into our future work.

6 Conclusions and Future Work

Moving from analysis of static collections to dynamic analysis of changing collections has opened important new applications and insights to our users; the paradigm shift closely echoes the nature of the problems, data, and tasks of the users.

Retaining interaction, analytic, and reasoning context over changing data remains a critical goal. The visual analysis system supports complex cognitive work and the continuous development of understanding and judgments about the available data. Most of the work after initial user feedback was to enhance state and context across increments, and the initial user goals for a dynamic experience have been met in this area.

We verified the importance to users of having a continuously available information resource, with minimal latency, that required minimal user interaction. The "free running" model adopted here achieves a continuous experience that a user may simply monitor while carrying out other tasks and may investigate on demand. One user likened this to a "my data on CNN" capability with an immediate access to visual analytics' quick ability to assess, explore, and evaluate data.

Perception of the "now" appears to have been achieved by this solution; additional tests across a broader range of users, data, and problems are ongoing. Part of our future work will include improving the ability to understand the transitional change over modest lengths of time. This is particularly important when disruptive change has occurred, for example the emergence of significant new themes. The current tool does not explicitly emphasize new themes nor visualize the transition. One challenge to accomplishing this task is that the layout and semantic meaning of the space will change, perhaps in a small way, perhaps in a large way. This means that methods requiring an anchored frame of reference, such as those presented in [8] do not apply in this case. Alternate ideas currently are being investigated for this need.

We are continuing to work on improving detailed understanding of change and change context. Some examples include the incorporation of event detection, derivation of the context surrounding themes, and improved temporal reasoning tools.

Acknowledgments

The authors gratefully acknowledge the guidance of Dr. Russ Rose, and the support of the PNNL IN-SPIRE team, especially Kris Cook, Barb Cheney, Wendy Cowley, Nick Cramer, Dennis Jen, Jim Thomas, Leigh Williams, Lisa Glasford, and Sharon Eaton.

References

[1] Conrad Albrecht-Buehler, Benjamin Watson, and David A. Shamma, "Visualizing Live Text Streams Using Motion and Temporal Pooling," *IEEE Computer Graphics and Applications*, 25(3):52-59, 2005

[2] James Allan, Stephen Harding, David Fisher, Alvaro Bolivar, Sergio Guzman-Lara, and Peter Amstutz, "Taking Topic Detection from Evaluation to Practice," *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, Big Island, Hawaii, January 3-6, 2005. On CD.

[3] Urik Brandes and Steven R. Corman, "Visual Unrolling of Network Evolution and the Analysis of Dynamic Discourse," *Proceedings of the IEEE Symposium on Information Visualization (InfoVis 2002)*, Boston, Massachusetts, pp. 145-152, 2002.

[4] Ed H. Chi and Stuart K. Card, "Sensemaking of Evolving Web Sites Using Visualization Spreadsheets," *Proceedings of Information Visualization '99*, San Francisco, California, pp. 19-25, 1999.

[5] D. Eichmann, M. Ruiz, P. Srinivasan, N. Street, C. Culy, and F. Menczer, "A Cluster-Based Approach to Tracking, Detection and Segmentation of Broadcast News," Proc. DARPA Broadcast News Workshop, Herndon, VA, February 28 - March 3, 1999, pp. 69-75.

[6] Jean Anne Fitzpatrick, James Reffell, and Moryma Aydelott, "BreakingStory: Visualizing Change in Online News," *Proceedings of CHI 2003, Poster Session*, Ft. Lauderdale, Florida, pp. 900-901, 2003.

[7] Elizabeth Hetzler and Alan Turner. "Analysis Experiences Using Information Visualization," *IEEE Computer Graphics and Applications*, 24(5):22-26, 2004.

[8] Lucy Nowell, Elizabeth Hetzler, and Ted Tanasse, "Change Blindness in Information Visualization: A Case Study," *Proceedings of Information Visualization 2001*, San Diego, California, pp. 15-22, 2001.

[9] Earl Rennison, "Galaxy of News: An Approach to Visualizing and Understanding Expansive News Landscapes," *Proceedings of the ACM Symposium on Use InterfaceSoftware and Technology (UIST '94)*, Marina del Rey, California, pp. 3-12, 1994.

[10] Pak Chung Wong, Harlan Foote, Dan Adams, Wendy Cowley, and Jim Thomas, "Dynamic Visualization of Transient Data Streams," *Proceedings of the IEEE Symposium on Information Visualization (InfoVis 2003)*, Seattle, Washington, pp. 97-104, 2003.